

Vrinda Store Annual Sales Report 2022

Month

Jan

Feb

Mar

Apr

Channel

Ajio

Amazon

Flipkart

Meesho

Myntra

Category

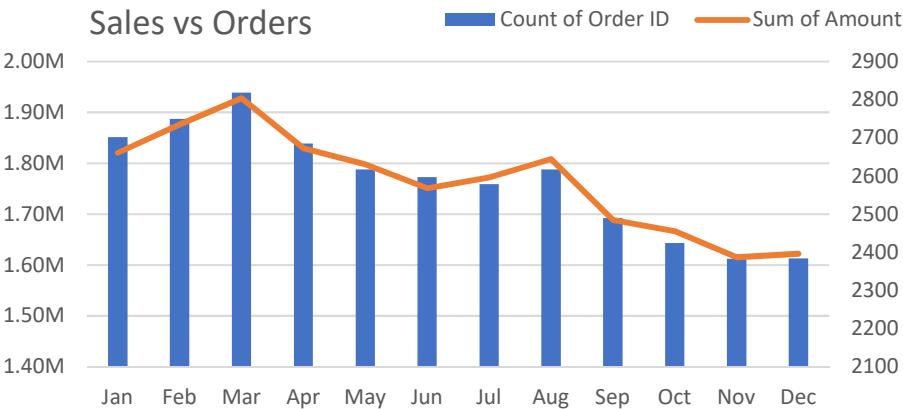
Saree

Set

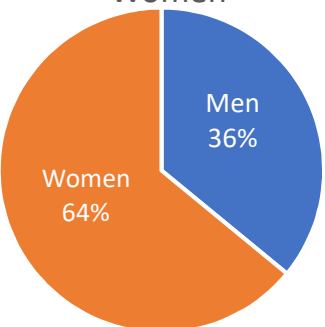
Top

Western Dress

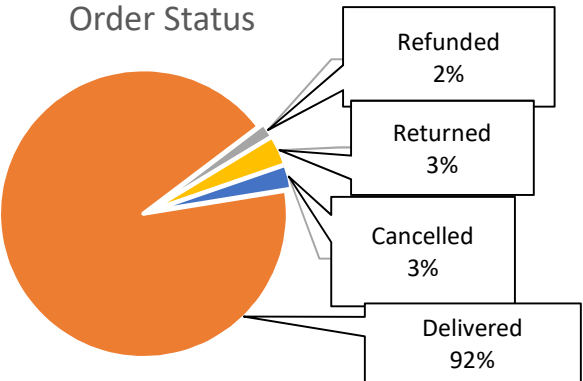
Sales vs Orders



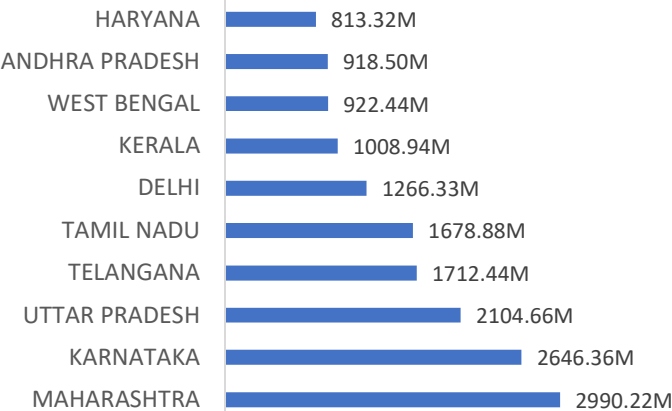
Purchased: Men vs Women



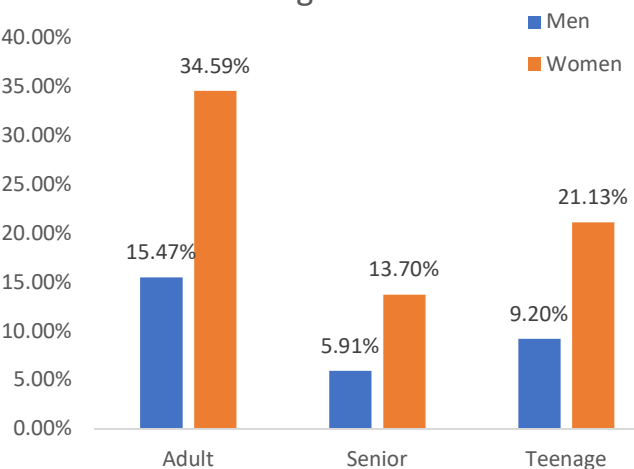
Order Status



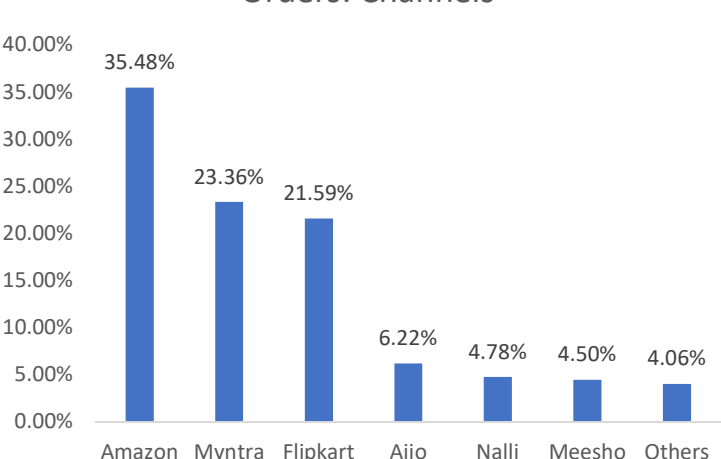
Top 10 States Contributing Sales



Orders: Age vs Gender



Orders: Channels



Vrinda Store Analysis

Project - 2022 Annual Report

```
In [87]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [57]: df= pd.read_excel("Vrinda Store Data Analysis.xlsx")
df.head(5)
```

Out[57]:

	index	Order ID	Cust ID	Gender	Age	Date	Status	Channel	SKU	Category	Si
0	1	171-1029312-3038738	1029312	Women	44	2022-12-04	Delivered	Myntra	JNE1233-BLUE-KR-031-XXL	kurta	X
1	2	405-2183842-2225946	2183842	Women	29	2022-12-04	Delivered	Ajio	SET414-KR-NP-L	Set	
2	3	171-1641533-8921966	1641533	Women	67	2022-12-04	Delivered	Myntra	SET261-KR-PP-S	Set	
3	4	404-7490807-6300351	7490807	Women	20	2022-12-04	Delivered	Amazon	SET110-KR-PP-M	Set	
4	5	403-9293516-4577154	9293516	Women	62	2022-12-04	Delivered	Myntra	JNE2294-KR-A-XXL	kurta	X

Data Cleaning

```
In [59]: df.isnull().sum()
```

```
Out[59]: index                0
Order ID                0
Cust ID                0
Gender                0
Age                0
Date                0
Status                0
Channel                0
SKU                0
Category                0
Size                0
Qty                0
currency                0
Amount                0
ship-city                0
ship-state                0
ship-postal-code        0
ship-country            0
B2B                    0
dtype: int64
```

```
In [60]: df.columns
```

```
Out[60]: Index(['index', 'Order ID', 'Cust ID', 'Gender', 'Age', 'Date', 'Status',
               'Channel ', 'SKU', 'Category', 'Size', 'Qty', 'currency', 'Amount',
               'ship-city', 'ship-state', 'ship-postal-code', 'ship-country', 'B2
               B'],
              dtype='object')
```

```
In [63]: df['Gender'].unique()
```

```
Out[63]: array(['Women', 'Men', 'W', 'M'], dtype=object)
```

Replacing 'M' and 'W'

```
In [67]: df['Gender'].replace({'M': 'Men', 'W': 'Women'}, inplace=True)
df['Gender'].unique()
```

```
Out[67]: array(['Women', 'Men'], dtype=object)
```

```
In [71]: contains_alphabets = df['Age'].apply(lambda x: any(char.isalpha() for char in x))

# Display the result
if contains_alphabets.any():
    print("The 'Age' column contains values with alphabets.")
else:
    print("The 'Age' column does not contain values with alphabets.")
```

The 'Age' column does not contain values with alphabets.

```
In [74]: contains_alphabets = df['Date'].apply(lambda x: any(char.isalpha() for char
# Display the result
if contains_alphabets.any():
    print("The 'Date' column contains values with alphabets.")
else:
    print("The 'Date' column does not contain values with alphabets.")
```

The 'Date' column does not contain values with alphabets.

```
In [75]: df['Status'].unique()
```

```
Out[75]: array(['Delivered', 'Refunded', 'Cancelled', 'Returned'], dtype=object)
```

```
In [76]: df['Size'].unique()
```

```
Out[76]: array(['XXL', 'L', 'S', 'M', 'XL', 'XS', '3XL', 'Free', '6XL', '4XL',
'5XL'], dtype=object)
```

```
In [77]: df['Qty'].unique()
```

```
Out[77]: array([1, 'One', 2, 4, 3, 'Two', 5], dtype=object)
```

Replace 'One' and 'Two'

```
In [79]: df['Qty'].replace({'One':1, 'Two':2}, inplace=True)
df['Qty'].unique()
```

```
Out[79]: array([1, 2, 4, 3, 5], dtype=int64)
```

```
In [80]: df['ship-country'].unique()
```

```
Out[80]: array(['IN'], dtype=object)
```

Data Preprocessing

Create new column for age group for senior, adult, teenage

```
In [83]: df['Age_Category'] = np.where(df['Age'] >= 50, 'Senior', np.where(df['Age']
```

```
In [84]: df['Age_Category']
```

```
Out[84]: 0      Adult
1      Teenage
2      Senior
3      Teenage
4      Senior
...
31042   Senior
31043   Adult
31044  Teenage
31045   Adult
31046   Adult
Name: Age_Category, Length: 31047, dtype: object
```

create new column name 'Month' month depends on Date column

```
In [86]: df['Month'] = df['Date'].dt.strftime('%b')
df['Month']
```

```
Out[86]: 0      Dec
1      Dec
2      Dec
3      Dec
4      Dec
...
31042   Jan
31043   Jan
31044   Jan
31045   Jan
31046   Jan
Name: Month, Length: 31047, dtype: object
```